보고서

1. 프로젝트 개요

1-1. 주제

- **무신사 스토어 리뷰 데이터 분석을 통한 의류 사이즈 재구성**

1-2. 배경

- 의류 구매 시 적합한 사이즈의 선택이 중요하며, 사이즈와 핏 문제는 주요 환불 원인 중 하나로, 이는 의류 구입에서 사이즈와 관련된 해결책이 필요함

- 참고 문헌

  ▼ "쇼핑몰 옷 사이즈 애매해 불편, 개선해달라" 청원

  "쇼핑몰 옷 사이즈 애매해 불편, 개선해달라" 청원
  표준의류사이즈표에도 불구하고 제각각인 쇼핑몰별 옷 사이즈로 인해 소비자들이 불편을 호소하고 있다. 이에 국내 의류 표준 규격 사이즈를 정해달라는 청원이 등장했다. 3일 청와대 국민청원 게시판에 '국내 의류 표준 규격 사이즈 관련 법을 제정해주십시오'라는 제목의 청원이 게시됐다.
  https://www.ekoreanews.co.kr/news/articleView.html?idxno=42138

  ⇒ "실제 한국소비자원에는 의류 사이즈 불만 관련 민원이 지속적으로 접수되는 상황이다. 인터넷 쇼핑몰에서 코트를 구입한 한 소비자는 착용해보니 쇼핑몰에서 설명한 사이즈와 달라 반품을 요구했다."

  ▼ 패션 상품의 반품의 비중은 전체 반품 상품 중 가장 높게 나타났는데 소비자의 45%가 사이즈, 핏, 컬러의 문제로 반품

  www.sfti.or.kr
  https://www.sfti.or.kr/pdf_files/pdf_file/25_3_02_280_290_23_663.pdf

  ⇒ "반품율에 대한 자세한 조사 결과는 "State of returns"(2022)에 따르면 2021년 온라인 쇼핑 전체 반품에서 의류가 차지하는 비중이 31%로 가장 높았고 반품의 이유에 대해서는 반품하는 소비자의 45%가 사이즈, 핏, 컬러의 문제로 반품한다는것이 1위를 차지하였다."

  ▼ 참고 프로젝트

  제 17회 보아즈(BOAZ) 빅데이터 컨퍼런스 - [SiZoAH] : 리뷰 기반 의류 사이즈 추천시스템
  제 17회 보아즈(BOAZ) 빅데이터 컨퍼런스 - [SiZoAH] : 리뷰 기반 의류 사이즈 추천시스템 - Download as a PDF or view online for free
  https://www.slideshare.net/slideshow/17-boaz-sizoah/255973431

1-3. 목적

- 옷 마다 실제 사이즈가 달라서 **리뷰 분석을 통해 실측 사이즈 정보를 재구성** 하고자 함

2. 프로젝트 수행 절차

1. 무신사 홈페이지에서 상의(후드티, 긴소매, 셔츠, 맨투맨) **상품 100개씩**, 각 상품의 **일반 리뷰 300개씩** 수집
2. 리뷰 데이터를 활용하여 **사전 구축**
3. 만들어진 사전을 활용하여 **리뷰 텍스트 분석** 진행

4. 리뷰 텍스트 분석을 통해 얻은 데이터를 **계산식에 활용해서 사이즈 재구성**

3. 데이터 수집

- 수집 방식
  - 무신사 스토어에서 제공하는 **API**를 활용하여 수집함

- 수집 기준
  - 남자 상의 (맨투맨, 셔츠, 후드티, 긴소매) **4개**의 카테고리에서 수집

  ▼ 카테고리별 **1년간 판매가 가장 많은 순서**로 정렬 후 **1위**부터 **100위**까지의 상품을 수집

  - 판매 높은순
    - **1년 단위 계절별** 옷을 모두 볼 수 있을 것이라 예상, **판매량이 많을수록 리뷰가 많음**
    - 다른 정렬 기준(무신사 추천순, 신상품(재입고순), 낮은가격순, 높은가격순, 할인율순, 후기순, 랭킹페이지)에 비해서 수집 정렬 기준으로 적합함
  - 다른 정렬 기준 기각 이유
    - **무신사 추천순** : 정렬 기준이 불분명하고 무신사의 방향성은 무신사 추천순의 최종 목표는 개인의 취향을 고려한 추천을 하고자 하는 것으로 보임 *(수집에 개인의 편향이 존재)*
    - **신상품(재입고순)** : 리뷰가 적어서 리뷰 데이터를 활용하기에 적합한 정렬 기준이 아님
    - **낮은가격순, 높은가격순** : 리뷰가 적은 상품들이 많이 존재, 의류 추천 시스템의 사용자가 최종 output을 좋아할 확률이 후기가 많은 상품을 수집해서 추천하는 것보다 낮음
    - **할인율순** : 할인 이벤트 하는 옷이 상단에 존재함
    - **후기순** : 무신사 자사 브랜드가 후기 이벤트로 인해 자사 브랜드의 상품이 다수 상단에 위치하여 데이터 다양성이 떨어짐
    - **랭킹** : 상품 매출이 기준의 큰 비중을 가져 비싼 제품일수록 유리함

  ▼ 한 상품에 대해 일반 리뷰를 유용한 순으로 **300개씩** 수집

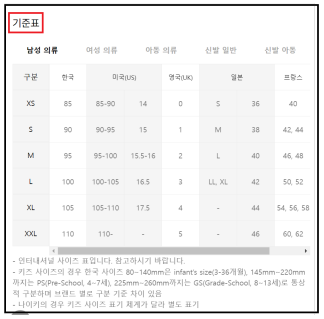
  - 무신사는 **사진리뷰 / 스타일리뷰 / 일반리뷰**로 리뷰를 구분하지만 사진 리뷰에는 일반 리뷰의 내용이 **중복**되는 내용이 있고**, 편향을 막고자 300개씩** 수집을 해야했기에 **가장 많은 개수가 있는 일반리뷰**로 채택함

  ▼ 리뷰 데이터에서 **유용한 순**으로 정렬하여 수집

  - 유용한 순은 **도움돼요의 수 기준**으로 정렬됨
    - 도움돼요 수가 많으면 **공감하는 사람이 많다**고 볼 수 있고 그 리뷰에 대한 **신뢰도가 높다고 판단**하여 리뷰 필터를 유용한순으로 채택함

  ▼ 각 상품마다 게시되어 있는 상품 **실측 사이즈 표** 수집

  - 실측 사이즈표가 없고 **기준표만 존재할 경우**, 기준표의 수치값이 사이즈 재구성에 적합하지않아 **수집 제외함**



- 수집 형태
  - `product_df` (상품의 정보 데이터)
    - shape: *(400, 6)*

Column	Information	Type	Describe
goodsNo	상품번호	int	
goodsName	상품명	object	
imageUrl	상품 이미지 링크	object	
relatedGoodsReviewScore	상품에 대한 총 평점	int	92 ~ 99
brandName	brand 한글명	object	
brandNameEng	brand 영문명	object	
cate	상품 카테고리	object	맨투맨 / 셔츠 / 후드티 / 긴소매

  - `review_df` (리뷰 데이터)
    - shape: *(117247, 9)*

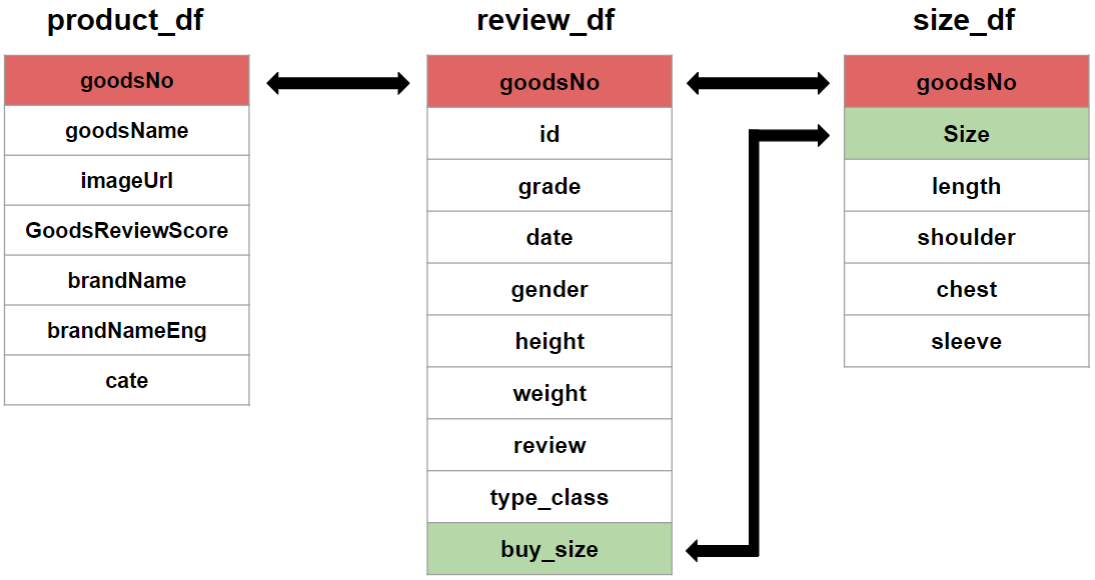
Column	Information	Type	Describe
id	사용자 ID	object	
grade	사용자 등급	int	0 ~ 8 (총 구매 금액의 따라 증가)
date	리뷰 작성 날짜	object	2017. 10. 10 ~ 2024. 06. 11
gender	성별	object	남 / 여 / 없음
height	키	int	0 ~ 220
weight	몸무게	int	0 ~ 150
review	상품에 대한 일반 리뷰	object	
type_class	사용자의 사이즈 평가 척도	object	작아요 / 보통이에요 / 커요 / 무응답
buy_size	구매 사이즈	object	S / M / L / XL / 1 / 2 / 3 / ...

  - `size_df` (실측 사이즈 표 데이터)
    - shape: *(1564, 6)*

Column	Information	Type	Describe
goodsNo	상품번호	int	
Size	상품 실측 사이즈	object	S / M / L / XL / 1 / 2 / 3 /...
length	총장길이	float	56 ~ 86.5
shoulder	어깨너비	float	43 ~ 78
chest	가슴단면	float	46 ~ 79.5
sleeve	소매길이	float	20 ~ 100

- 수집 기간
  - **2024. 06. 11** 데이터 수집
    - **실시간으로 순위가 변경**되고 있기 때문에 프로젝트에 사용할 데이터 수집은 **하루 안에 진행**했고, 추후에 데이터 수집을 다시 진행하게 된다면 상품 목록이 바뀔 수 있음

- 데이터 병합
  - `goodsNo`를 **key값**으로 하나의 데이터 프레임 병합
    1. `review_df`와 `size_df`를 병합하여 `review_size_df` 생성함
       - `goodsNo`를 **key값**으로 `review_df`의 `buy_size`와 `size_df`의 `Size`을 매칭시켜 1차적 병합 진행함
    2. `product_df`와 `review_size_df`를 병합하여 최종 데이터 프레임 생성
       - `goodsNo`를 **key값**으로 두 데이터 프레임을 병합함



- 최종 병합 데이터 프레임
  - shape: *(95400, 19)*
    - 상품 개수(318) * 리뷰 데이터 개수(300)

	goodsNo	brandNameEng	goodsName	imageUrl	reviewScore	review	typeClass	cate	size	length	shoulder	chest	sleeve	id	date	grade	gender	height	weight
0	2272830	SPAO	(시티보이) 오버핏 옥스포드 셔츠 _SPYWE23C51	https://image.msscdn.net/thumbnails/images/goo...	96	역사 스파오 셔츠는 그냥 기본으로 하나씩은 들고있어야 합니다! 다른 색깔들도 색감 ...	커요	shirt	XXL	80.0	65.0	69.0	59.0	준준준0101	2023.09.27	6	남성	174	75
1	2272830	SPAO	(시티보이) 오버핏 옥스포드 셔츠 _SPYWE23C51	https://image.msscdn.net/thumbnails/images/goo...	96	이미지 체인지를 조금 해보고싶어서... 항상 스트라한 룩만 입다가 남친죽임어쩔려고 ...	보통이에요	shirt	L	77.0	62.0	64.0	56.5	tae_tuyu	2023.09.05	5	남성	174	52
2	2272830	SPAO	(시티보이) 오버핏 옥스포드 셔츠 _SPYWE23C51	https://image.msscdn.net/thumbnails/images/goo...	96	스파오 셔츠는 색상별로 가서 이번이 3번째입니다! 저는 오버한 핏을 원했어서 라지...	커요	shirt	L	77.0	62.0	64.0	56.5	모호한대구트백	2023.10.07	5	남성	170	68
3	2272830	SPAO	(시티보이) 오버핏 옥스포드 셔츠 _SPYWE23C51	https://image.msscdn.net/thumbnails/images/goo...	96	색깔도 예쁘게 뽑았고 깔끔하고 단정하게 입기 좋아요!	보통이에요	shirt	XL	79.0	63.5	66.5	58.0	대세는남산아울렛	2024.03.16	4	미상	0	0
4	2272830	SPAO	(시티보이) 오버핏 옥스포드 셔츠 _SPYWE23C51	https://image.msscdn.net/thumbnails/images/goo...	96	안녕하세요.후기를 남기기에 앞서 저는 키가 175cm, 몸무게 80kg인 30대 ...	보통이에요	shirt	XL	79.0	63.5	66.5	58.0	성으뜸	2023.11.19	4	남성	175	80

4. 데이터 전처리

4-1. 데이터 전처리

- 수집 **기준에 맞지 않는** 데이터 제거
  - 작성된 리뷰의 개수 자체가 **300개가 되지 않는** 데이터 **8,347개** 제거함
  - ▼ 무신사에서 **실측 사이즈표를 제공하지 않는** 데이터 **1,200개** 제거함
    - 무신사는 상품에 대한 **실측 사이표와 기준 사이즈표를 동시에 제공**하고 있지만 **특정 상품에 대해서는 기준 사이즈표만 제공** 하고 있어 실측 사이즈표를 수집 하지 못한 상품 제거함
  - 위 **두개의 문제를 동시**에 가지고 있는 데이터 **486개** 제거함
  - 위 두개의 문제를 가지고 있지 않지만 **전처리 및 병합 과정**에서 `buy_size` / `Size` 를 매칭시키는 과정에서 매칭이 안되어 리뷰 개수 300개 미만인 데이터 **11,200개** 제거함

- `review_df`의 `buy_size`와 `size_df`의 `Size` 를 병합하기 위한 사전 전처리
  - ▼ 데이터 통일을 위해 **정규 표현식**을 활용하여 **한글 / 숫자 / 특수문자** 제거

```
review_df["buy_size"] = review_df["buy_size"].apply(lambda x: re.sub(r'[가-힣()`
size_df["Size"] = size_df["Size"].apply(lambda x: re.sub(r'[가-힣()\[\]{}\/]',
```

  - ▼ 사이즈 정보가 아닌 **불필요한 단어 제거**

```
def clean_list(lst):
    # 빈 문자열과 'NO'를 제거
    cleaned_list = [item for item in lst if item not in ['', 'NO']]

    # 특정 문자열 조합을 결합
    combined_list = []
    i = 0
    while i < len(cleaned_list):
        if i < len(cleaned_list) - 1 and cleaned_list[i] == "EXTRA" and cleaned
            combined_list.append("EXTRALARGE")
            i += 2  # 두 항목을 하나로 합쳤으므로 두 인덱스를 건너뜀
        else:
            combined_list.append(cleaned_list[i])
            i += 1  # 다음 인덱스로 이동

    return combined_list
```

  - ▼ **표기는 다르나 사이즈는 동일**한 경우 중복 제거
    - 위의 과정을 진행하면 ***논기모_XS*** 가 ***_XS***로 처리되고 중복값이 발생하기 때문에 중복 제거함

논기모_XS	66	56.5	56.5	56.1
논기모_S	68	59	59	57.4
논기모_M	70	61.5	61.5	58.7
논기모_L	72	64	64	60
논기모_XS (후드)	66	56.5	56.5	56.1
논기모_S (후드)	68	59	59	57.4
논기모_M (후드)	70	61.5	61.5	58.7
논기모_L (후드)	72	64	64	60

▼ 병합된 데이터끼리 **딕셔너리 매핑**을 통해 **통일화**

- ex) *XXL*100 - *XXL* / *XL*105 - 100
- 위와 같은 데이터의 구성으로 **같은 값을 나타내지만 데이터의 형식이 달라** 병합하지 못함
- 값의 **unique**를 확인하여 **딕셔너리로 매핑** 처리함
- 특정 사이즈 표기의 경우 브랜드만의 **사이즈 표기로 파악이 불가능한** 사이즈는 제거함
  - ex) 1 / 2 / 3 으로 표기

```
# 사이즈 통일 딕셔너리

size_trans_dict = {
    "S" : "S", "M" : "M", "L" : "L", "XL" : "XL", "XXL110" : "XXL", "L100" :
    "S090" : "S", "M095" : "M", "XS085" : "XS", "2XL110115" : "XXL", "L10010
    "M95100" : "M", "1M" : "M", "3XL" : "XXXL", "4895" : "095", "50100" : "1
    "2L" : "L", "Large" : "L", "Small" : "S", "Xlarge" : "XL", "Medium" : "M
    "M1" : "M", "S0" : "S", "L2" : "L", "M95" : "M", "LARGE" : "L", "SMALL"
    "XSWOMEN" : "XS", 'FREE' : "FREE", 'SWomen' : "S", 'Free' : "FREE",'4XL'
    # 'M100' : "M", 'L105' : "L", 'S95' : "S", 'XL110' : "XL"
    #'085', '095', '105', '090', '100', '110', "1", "2", "3", '4', 'LL', '3L
}
```

- 사용자 `ID` / `date` / `grade` 각각 **단일 컬럼**으로 분리
  - 탈퇴회원은 `ID`를 **탈퇴회원**으로 구분
    - 총 **51개** 데이터
    - 사용자 `ID` 컬럼을 데이터의 고유값으로 구분하는데 용도 이외의 다른 컬럼의 정보는 가지고 있기 때문에 제거 하지 않음

- `height` / `weight` / `gender` 각각 단일 컬럼으로 분리
  - 사용자의 신체 정보가 현재 프로젝트 목적에 필요하지는 않지만 추후에 **사용자의 교환 사이즈나 반품 사이즈 데이터 정보와 결합**하면 추가적인 분석이 가능하여 제거하지 않음
  - `gender` 컬럼 **37,810개 Null값**에 대하여 **"미상"**으로 변경

- `height` / `weight` 컬럼 **37,786개 Null값**에 대하여 **"0"** 으로 변경

- `typeClass` 를 **target**으로 사용하기 위한 전처리
  - "사이즈 보통이에요"를 "보통이에요"의 형식으로 **"사이즈" 단어 삭제**
  - `typeClass`의 대한 결측치 처리
    - 개인이 리뷰를 작성할 때 옷의 전반적인 평가를 선택할 수 있으며 **'작아요', '보통이에요', '커요'** 3가지의 선택지가 있음
    - 고객이 **옷 전반적인 사이즈에 대해 의견**을 제시한 것으로 판단하여, 추후 사이즈 재정비 방향이 맞는지 확인하는 용도로의 방향성으로 잡음
    - `typeClass` 전체 비율 시각화
      - **Null**값의 비율 : **0.52%**
      - **Null**일 경우, 고객이 구매한 의류 상품의 사이즈에 대해 **특정한 의견을 제시하지 않았음**으로 판단해 **'보통이에요'** 처리함

- `goodsNo` / `id` / `review` 를 기준으로 중복 제거
  - **동일한 상품을 여러번 구매**하여 동일한 리뷰 내용을 작성한 데이터 **800개 중복 제거**

- `review` 데이터 전처리

  ▼ **리뷰 데이터 특성을 고려**하여 분석 목적에 맞추어 **특수문자 / 반복문자 / 단일 모음 자음 / 외국어 데이터 제거**

  - 리뷰 데이터의 경우 형식이 맞춰져 있지 않고 다양한 형태로 구성되어 있기 때문에 프로젝트 목적에 맞는 데이터를 분석하고 추출하기 위해 데이터 클리닝을 진행함

```
def clean_text(text):

    # 특수문자 제거 (한글, 영어, 공백)
    text = re.sub(r'[^가-힣\s]', '', text)

    # 반복 문자 제거 (예: "!!" -> "!")
    text = re.sub(r'(.)\1+', r'\1', text)

    # 단일 자음 및 모음 제거
    single_korean_letters = r'\b[ㄱ-ㅎㅏ-ㅣ]\b'
    text = re.sub(single_korean_letters, '', text)

    # 외국어만 있는 텍스트 제거 (한국어가 포함되지 않은 텍스트)
    if not re.search(r'[가-힣]', text):
        return ''

    return text
```

원본 review	정제 review
안녕하세요.후기를 남기기에 앞서 저는 키가 175cm, 몸무게 80kg 인 30대 남성입니다.이 셔츠는 정말이지 최고입니다.솔직후기 적겠습니다.저를 믿고 구매하셔도 됩니다!! ✅ ... (생략)	안녕하세요후기를 남기에 앞서 저는 키가 몸무게 인 대 남성입니다이 셔츠는 정말이지 최고입니다솔직후기 적겠습니다저를 믿고 구매하셔도 됩니다 ... (생략)

  - 모든 문장이 외국어로만 구성되어 있는 **리뷰 41개** 제거함

- 최종 전처리 후 최종 데이터 프레임
  - shape: *(94559, 19)*

4-2. 활용 라이브러리 등 기술적 요소

▼ 라이브러리

- requests
- BeautifulSoup
- numpy
- pandas
- re
- matplotlib.pyplot
- seaborn
- json
- okt
- mecab
- kiwi
- Counter
- WordCloud
- Word2Vec
- sklearn

5. 데이터 분석

5-1. 리뷰 데이터 활용 사전 구축

- **의류 사이즈를 재구성**하는 프로젝트의 목적에 따라 디자인이나 컬러의 내용과 사이즈의 내용을 같이 가지고 있는 정보라도 **"사이즈"와 관련된 리뷰의 내용만 활용해야함**

실제 데이터 문장	필요한 데이터 문장
옷이 너무 이쁘고 마음에 들어요. 하지만 총장과 소매가 길고, 가슴너비가 작아요	총장과 소매가 길고, 가슴너비가 작아요

- 필요한 내용에 대하여 **핵심 키워드를 설정**
  - 키워드를 설정하여 필요한 내용 즉, **사이즈에 관련된 정보**만 활용
  - **전체 / 총장 / 어깨 / 가슴 / 소매**
    - 무신사 제공 실측 사이즈 표를 바탕으로 **4가지 부위와 전체 사이즈** 관련 키워드 선정
  - **크다 / 작다**
    - **사이즈 크기 대한 표현**으로 크다, 길다 / 작다, 짧다 등 크기 관련 키워드 선정

- 키워드를 중심으로 **공백 기준 단어 빈도** 확인

  ▼ 키워드에 대해 **빈도 Top 100**과 **키워드의 형태**들을 `WordCloud`로 확인

- 기장

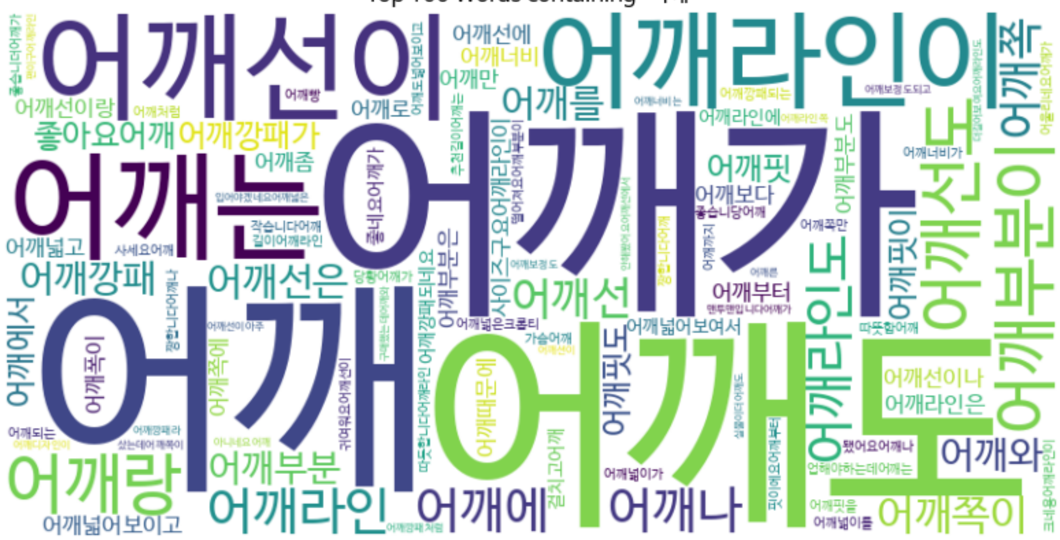

Top 100 Words containing '기장'
기장도
기장
기장이
기장은
기장감이
기장감
기장에
총기장은
기장이라
기장을
기장이랑
기장감도
기장이나
총기장이
기장감은
팔기장이
소매기장이
기장감이랑
기장이고
기장으로
기장입니다
기장이에요
기장과
팔기장도
팔기장은
기장만
뒷기장이
기장이며
기장감에

- 가슴


Top 100 Words containing '가슴'
가슴둘레
가슴
가슴이
가슴에
가슴은
가슴둘레가
가슴쪽이
가슴쪽에
가슴폭이
가슴쪽
가슴쪽은
가슴팍에
가슴부분이
가슴어깨
가슴팍
가슴단면
가슴품
가슴로고에
가슴팍이
가슴도
가슴하고
가슴과
가슴을
딱이네요어깨가슴소매기장
많은데가슴과
선택을잘한거같아요가슴에
같아요가슴팍
검정색인데가슴너비가

- 어깨

Top 100 Words containing '어깨'



- 소매

Top 100 Words containing '소매'

- 크다 / 길다

Top 100 Words containing '크다'

- 작다 / 짧다

  - 키워드를 중심으로 다양한 형태의 단어들이 존재함을 확인
    - 크다: 큰 / 큰데 / 커서 / 커요 / 커용
    - 작다: 작은데 / 작아서 / 작은 / 작아요

- **비슷한 유형의 프로젝트를 참고**하여 사전 구성의 기반을 잡고, 현재 가지고 있는 데이터를 중심으로 **사전을 재구성**
  - `Word2Vec`을 통해 키워드별 단어의 **유사도를** 통해 비슷한 단어로 사전 재구성

    ▼ 기존 키워드 사전을 `Word2Vec`을 활용하여 단어와 유사한 단어를 찾고 **상위 30개**의 단어를 기준으로 데이터에 맞게끔 단어들을 **추가하거나 제거함**

```
okt_word2vec = Word2Vec(okt_tokenized_reviews, vector_size = 200, window = 1
okt_word2vec.wv.most_similar("총장", topn = 30)
```

```
1 okt_word2vec.wv.most_similar("총장", topn=30)

('폼', 0.8637468814849854),
('통', 0.8448014259338379),
('몸통', 0.8392444252967834),
('기장', 0.8363900184631348),
('품은', 0.8043614029884338),
('길이', 0.800937294960022),
('팔이', 0.7963797450065613),
('소매', 0.7929900884628296),
('폭', 0.7873339653015137),
('허리', 0.7694438695907593),
```

- 재구성된 사전 샘플

	참고한 키워드 사전	유사도를 통해 재구성한 키워드 사전
전체 핏	사이즈하 / 사이즈감 / 요핏 / 싸이즈 / 핏을	사이즈 / 핏 / 크기 / 전체 / 상체 / 체구 / 골격
총장	밑기장 / 길이 / 총장 / 궁디 / 엉덩이	기장 / 총장 / 세로 / 밑단 / 아랫쪽 / 하단
어깨	어깨핏 / 어깨 / 너비 / 광배 / 어좁	어깨핏 / 어깨 / 어깡 / 광배 / 어좁
가슴	가슴 / 바디 / 가로/ 넓이 / 통 / 몸통	가슴 / 바디 / 통 / 몸집 / 가슴팍 / 몸통
소매	소매 / 팔도 / 손목/ 손 / 팔다리 / 당팔 / 팔길이	소매 / 팔 / 손목 / 손 / 팔목 / 팔길이 / 팔기장
크다	커서 / 큼 / 넉넉하게 / 넉넉한 / 오버핏이 / 벙벙하다	커서 / 큼 / 크구요 / 길구요 / 박시하게 / 큽니다 / 오버핏
작다	크롭된 / 짧아요 / 짧습니다 / 작아요 / 숏한 / 크롭한	작습니다 / 짧아요 / 짧습니다 / 작아요 / 작았으면 / 작음

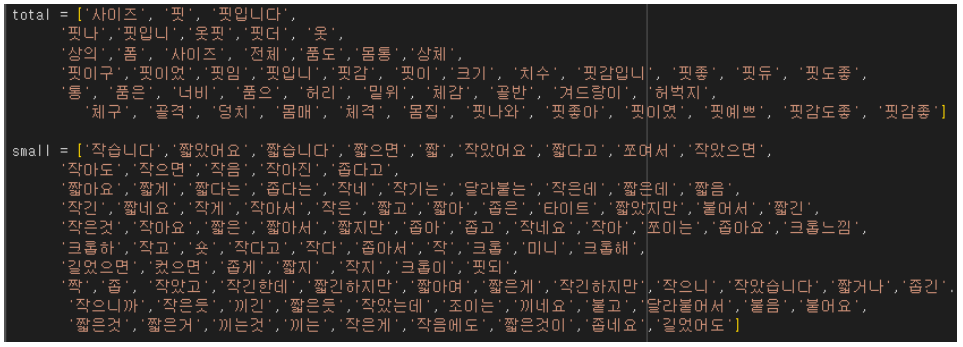
5-2. 사전 기반 리뷰 분석

- **length, shoulder, chest, sleeve** 사전 → **big, small** 판단
  - 방식 : 리뷰 텍스트에서 **사이즈와 관련된 내용만이 본 프로젝트의 목적에 부합**하다고 판단함
    1. `okt` 형태소 분석기를 활용하여, **'Verb' (동사) 기준으로 ',' 삽입**함
    2. ','를 기준으로 분리한 문장을 `cate_review` 각 행에 넣음
    3. 각각의 텍스트에서 **신체 사이즈 관련 사전[length, shoulder, chest, sleeve]에 있는 단어의 존재 여부를 파악**함
    4. 같은 텍스트 내에 **크기 판단 사전[big, small]**에 있는 단어의 존재 여부를 파악함
    5. 포함되어 있다면 **1**, 포함되지 않는다면 **0**을 부여함
  - ',' 기준으로 문장을 각각의 행으로 분리한 데이터
    - `reveiw` : 원본 리뷰 텍스트
    - `cate_review` : ','로 분리된 문장을 각각 넣음

review	cate_review
색깔 도 예쁘게 빠졌고, 깔끔하고 단정하게 입기, 좋아요	색깔 도 예쁘게 빠졌고
색깔 도 예쁘게 빠졌고, 깔끔하고 단정하게 입기, 좋아요	깔끔하고 단정하게 입기
색깔 도 예쁘게 빠졌고, 깔끔하고 단정하게 입기, 좋아요	좋아요

    - 기존 데이터 shape : *(94559, 19)*
    - 문장 분리 후 데이터 shape : *(329271, 30)*
    - `cate_review` **Null** 값 drop 데이터
      - `new_df`
        - shape : *(305351, 20)*

- 카테고리 별 Dict 생성
  - 사전 리스트 형태



```
total = ['사이즈', '핏', '핏입니다',
      '핏나','핏입니','옷핏','핏더', '옷',
      '상의','폼', '사이즈', '전체','품도','몸통','상체',
      '핏이구','핏이었','핏임','핏입니','핏감', '핏이','크기', '치수', '핏감입니', '핏좋', '핏듀', '핏도좋',
      '통', '품은', '너비', '품으', '허리', '밑위', '체감', '골반', '겨드랑이', '허벅지',
        '체구', '골격', '덩치', '몸매', '체격', '몸집', '핏나와', '핏좋아', '핏이였', '핏예쁘', '핏감도좋', '핏감좋']

small = ['작습니다','짧았어요','짧습니다','짧으면','짧','작았어요','짧다고','쪼여서','작았으면',
      '작아도','작으면','작음','작아진','좁다고',
      '짧아요','짧게','짧다는','좁다는','작네','작기는','달라붙는','작은데','짧은데','짧음',
      '작긴','짧네요','작게','작아서','작은','짧고','짧아','좁은','타이트','짧았지만','붙어서','짧긴',
      '작은것','작아요','짧은','짧아서','짧지만','좁아','좁고','작네요','작아','쪼이는','좁아요','크롭느낌',
      '크롭하','작고','숏','작다고','작다','좁아서','작','크롭','미니','크롭해',
      '길었으면','컸으면','좁게','짧지','작지','크롭이','핏되',
      '짝','좁', '작았고','작긴한데','짧긴하지만','짧아여','짧은게','작긴하지만','작으니','작았습니다','짧거나','좁긴',
       '작으니까','작은듯','끼긴','짧은듯','작았는데','조이는','끼네요','볼고','달라붙어서','붙음','붙어요',
       '짧은것','짧은거','끼는것','끼는','작은게','작음에도','짧은것이','좁네요','길었어도']
```

- 사진과 같은 사전 리스트에서 **cate(total/length/shoulder/chest/sleeve) + small/big** 리스트를 합친 **딕셔너리**를 생성함
  - **리스트 내 중복 단어를 제거**하여, 리스트 안에 **단어가 딕셔너리에 한 개 씩만 들어가도록** 함
  - 총 **5개**의 **cate(total/length/shoulder/chest/sleeve)** 딕셔너리 생성

  ▼ 딕셔너리 형태

```
# 예시
# categories_total

{'Total': ['핏이라서',
  '품',
  '좋아요핏',
  '핏처럼',
  '전체', ...,
 'Small': ['작았으면',
  '사이즈업',
  '짧네요',
  '짧았지만',
  '좁고'] ...,
  'Big': ['아방방한',
  '흘러내리는',,
  '크지', ...
  '루즈핏이',
  '넉넉합니다']}
```

- Dict내 단어 별 키워드 매칭
  - 신체 사이즈 관련 단어별로 생성한 Dict를 통해, 각각의 단어를 **key 값**으로 두고, 카테고리를 **value값**으로 두는 형태의 딕셔너리 형태로 변환
  - 총 5개의 **wordSizDictotal/length/shoulder/chest/sleeve 딕셔너리** 생성

  ▼ 딕셔너리 형태

```
{'치수': 'Total',
'핏좋': 'Total',
'허리': 'Total',
    ...
 '짧았어요': 'Small',
 '짧습니다': 'Small',
 '짧으면': 'Small'
    ...
```

```
'커도': 'Big',
'남아': 'Big',
'롱': 'Big'}
```

- 데이터 프레임 초기 값 설정
  - 카테고리 별 데이터 프레임 형식
    - row : 리뷰의 개수 → **30,5351**
    - column : `Total` / `Length` / `Shoulder` / `Chest` / `Sleeve` , `Small` , `Big`
      - 컬럼명은 Dict 내 단어 별 키워드 매칭 단계에서, value 값으로 지정된 명칭을 사용함함
  - 초기 값 0.0 데이터 프레임 생성
    - 리뷰의 개수가 행의 개수이고, 카테고리 별로 컬럼 명을 가지는 `sizedictCheck_total` / `length` / `shoulder` / `chest` / `sleeve` **5개**의 데이터 프레임을 형성함

index	Total	Small	Big
0	0.0	0.0	0.0
1	0.0	0.0	0.0
2	0.0	0.0	0.0
3	0.0	0.0	0.0
4	0.0	0.0	0.0
5	0.0	0.0	0.0
6	0.0	0.0	3.0
7	0.0	0.0	0.0
8	1.0	0.0	0.0
9	0.0	0.0	0.0
10	0.0	0.0	0.0
11	0.0	0.0	0.0
12	0.0	0.0	0.0
13	0.0	0.0	0.0

▼ 코드

```
# total 데이터 프레임 컬럼명
sizes_total = ['Total', 'Small', 'Big']
# length 데이터 프레임 컬럼명
sizes_length = ['Length', 'Small', 'Big']
# shoulder 데이터 프레임 컬럼명
sizes_shoulder = ['Shoulder', 'Small', 'Big']
# chest 데이터 프레임 컬럼명
sizes_chest = ['Chest', 'Small', 'Big']
# sleeve 데이터 프레임 컬럼
sizes_sleeve = ['Sleeve', 'Small', 'Big']
```

```
# index로 넣을 reviewIndex 빈 리스트 생성
reviewIndex = []
# new_df_catereviewlist의 리뷰 개수만큼 반복
for i in range(0,len(new_df_catereviewlist)):
  reviewIndex.append(str(i))
# 0으로 이루어진 단어 행렬 생성 (초기값)
sizedictCheck_total = pd.DataFrame(0.0, index=reviewIndex, columns=sizes_tot
print(sizedictCheck_total)
```

- 사전 등장 단어 count
  - count 방식
    - 리뷰 텍스트에서 **사전 딕셔너리에 있는 단어가 등장한 횟수를 카운트**하는 방식임
      - ex) index 2454 : Total 사전 단어 **6번** 등장, Small 사전 단어 **1번** 등장, Big 사전 단어 **2번** 등장 의미임

index	Total ▼	Small	Big
2454	6.0	1.0	2.0
668	4.0	0.0	2.0
12034	4.0	1.0	0.0

    - 이와 같은 방식으로 `df_sizedict_check_total` / `length` / `shoulder` / `chest` / `sleeve` **5개**의 데이터 프레임 형성함

- 카테고리 별 Small, Big 판단
  - 리뷰 텍스트에서 **신체 사이즈 관련 사전의 단어가 등장**하고, **크기 판단 사전의 단어가 등장**할 때, 이를 **리뷰 텍스트에서 '사이즈'와 관련된 리뷰라고 판단**함
  - 단어가 등장했음을 판단하는 지표 : `column` **> 0**
    - 신체 사이즈 관련 컬럼(`Total` , `Length` , `Shoulder` , `Chest` , `Sleeve`) 값이 **0 초과**할 때
      - 크기 판단 관련 컬럼(`Big` , `Small`) 중 `Big` 에서 **0을 초과**한다면, `total_big` 에 **1**을 부여하고, `Small` 에서 **0을 초과**한다면, `total_small` 에 **1을 부여**함
    - 신체 사이즈 관련 컬럼(`Total` , `Length` , `Shoulder` , `Chest` , `Sleeve`) 값이 **0 이하** 일 때
      - 크기 판단 관련 컬럼 값에 상관 없이, `total_big` , `total_small` 에 **0을 부여**함
  - 이와 같은 방식으로 `df_sizedict_check_total` / `length` / `shoulder` / `chest` / `sleeve` **5개**의 데이터 프레임 형성함

- 원본 데이터, 카테고리 병합 데이터 최종 병합
  - `new_df, df_sizedict_check_total` / `length` / `shoulder` / `chest` / `sleeve` 병합함
    - `merged_df`
      - shape : `(305351, 30)`

- 최종 병합
  - `merged_df` 데이터에서 `review` , `id` , `goodsNo` 기준으로 행을 합치고, `cate_review` 열 삭제
    - **',' 를 기준으로 리뷰 텍스트를 분리**하고, 이를 `cate_review` 에 분리된 텍스트를 각 행으로 넣었을 때, 다른 column의 값들은 모두 중복된 값을 가짐
    - 따라서 행의 개수가 **94,559개**에서 **329,271개**로 증가하였음
    - 이를 기존 데이터 프레임 형식으로 만들고자, `review` , `id` , `goodsNo` 를 기준으로 행을 합침
      - `result_df`
        - 텍스트 정제 작업 결과로 진행 된 리뷰 텍스트 데이터에서 `groupby` 로 병합을 진행했을 때, 동일하게 처리되는 리뷰 데이터가 존재하여, 최종적으로 **34개**의 row가 제거됨
        - shape : *(94525, 29)*

review	가격 대비 가성 비 가 너무 좋아요 핏 도 벌룬핏 으로 아주 맘에듭니 다
id	7573737
goodsNo	3050501
brandNameEng	JELEVE
goodsName	[2pack] 루즈핏 롱슬리브
imageUrl	https://image.msscdn.net/thumbnails/images/goo...
reviewScore	97
typeClass	커요
cate	longs
size	3
length	75.0
shoulder	65.0
chest	70.0
sleeve	57.0
date	2024.02.06
grade	LV.4
gender	남성
height	180
weight	80
total_big	1
total_small	0
length_big	0
length_small	0
shoulder_big	0
shoulder_small	0
chest_big	0
chest_small	0
sleeve_big	0
sleeve_small	0

5-3. 리뷰 분석 데이터 분류 정확도 평가

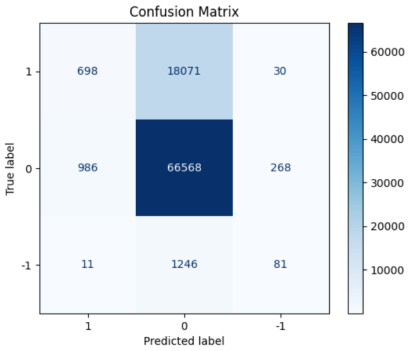
- 사전 기반 리뷰 분석이 끝난 데이터 사용
  - `length`, `shoulder`, `chest`, `sleeve` 정보가 **0**인 데이터 제거함
    - 상품 **1**개의 리뷰 **1**개 제거함
  - **FREE** 사이즈 제거
    - 상품 **3**개의 리뷰 **899**개, 상품 **1**개의 리뷰 **7**개 제거함(총 리뷰 수 **906**개 제거함)
  - `shoulder` 값이 **Null**인 경우 제거
    - 상품 **22**개의 **5,659**개의 리뷰 제거함
    - 어깨선이 존재하지않아 `sleeve` 값에 `shoulder` 값이 포함되어 있어서 어깨너비에 관련한 리뷰를 적용할 수 없다고 판단하여 삭제함
  - 컬럼 수정 내용
    - 앞서 리뷰 분석을 통해 얻은 데이터는 **각 부위별로 크고 작은 표현의 포함의 여부**를 각각 나타냄
      - `total_big`, `total_small`, `length_big`, `length_small`, `shoulder_big`,...
    - 각각 나타낸 정보를 하나의 부위에 크고 작음을 **한번에 분류**하기 위해서 컬럼을 뺄셈으로 합쳐 **새로운 파생변수**를 만듦
      - `total_b_s` : **전체 사이즈**가 **크다|보통이다|작다(1|0|-1)**를 분류하는 컬럼
        - `total_b_s` = `total_big` - `total_small`
      - `length_b_s` : **총장**이 **크다|보통이다|작다(1|0|-1)**를 분류하는 컬럼
        - `length_b_s` = `length_big` - `length_small`
      - `shoulder_b_s` : **어깨너비**가 **크다|보통이다|작다(1|0|-1)**를 분류하는 컬럼
        - `shoulder_b_s` = `shoulder_big` - `shoulder_small`
      - `chest_b_s` : **가슴둘레**가 **크다|보통이다|작다(1|0|-1)**를 분류하는 컬럼
        - `chest_b_s` = `chest_big` - `chest_small`
      - `sleeve_b_s` : **소매길이**가 **크다|보통이다|작다(1|0|-1)**를 분류하는 컬럼
        - `sleeve_b_s` = `sleeve_big` - `sleeve_small`

      ▼ 문제점

      - 하지만, 하나의 리뷰에서 하나의 부위에 대해 **동시에** 크고 작음이 존재하는 모순되는 정보가 발생했지만(245개 리뷰), **데이터 수정 작업 없이** 위의 뺄셈을 적용함으로써 그 리뷰의 해당 부위는 0(보통)값으로

계산되어 사용함

- 리뷰 분석 데이터 분류 정확도 평가
  - **하나의 리뷰**에 대해 `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s` 에 해당하는 값들을 다 더해서 새로운 파생변수인 `sum_b_s(전체적인 사이즈 평가|-4~4사이의 값)`에 값 저장
    - `sum_b_s` 값이 **0보다 크다**면 그 리뷰는 그 상품을 최종적으로 **크다**고 평가한 것이라 판단해서 새로운 파생변수인 `predicted_typeClass(전체적인 사이즈 평가|-1~1사이의 값)`에 **1**값 저장함
    - `sum_b_s` 값이 **0이면** 그 리뷰는 그 상품을 최종적으로 **적당하다**고 평가한 것이라 판단해서 새로운 파생변수인 `predicted_typeClass`에 **0**값 저장함
    - `sum_b_s` 값이 **0보다 작다면** 그 리뷰는 그 상품을 최종적으로 **작다**고 평가한 것이라 판단해서 새로운 파생변수인 `predicted_typeClass`에 **-1**값 저장함
  - `predicted_typeClass`과 리뷰를 입력한 사람이 남긴 `typeClass` 값을 비교하여, **리뷰를 올바르게 분석했는지 정확도**를 계산함
    - `typeClass` 는 리뷰를 남긴 사용자가 텍스트로 남긴 상품의 정보가 아닌, **사이즈 평가로 3가지의 선택지를 선택**함으로써 크고 작음에 대한 정보를 남긴 데이터라서, 리뷰 분석을 올바르게 분석하였는지 판단하기위한 **타겟값**으로 사용함
  - 결과
    - **Accuarcy** 값은 텍스트로 구성된 리뷰 분석이 선택값으로 구성된 리뷰와 **동일한 값을 가지는지** 정확도를 나타냄
    - **Accuracy : 0.77**
  - 혼동행렬 시각화
    - `typeClass`는 '**큰**' 반면, `predicted_typeClass`는 '**보통**'으로 분석되는 케이스가 많아 정확도가 떨어짐을 알 수 있음



5-4. 리뷰 분석 데이터 기반 사이즈 재구성

- 사이즈 재구성의 목표
  - **리뷰 분석을 통해 얻은 비율값**과 **상품의 각 부위별 사이즈 간격의 값**을 구해서, 그 두개의 값을 곱한 후 기존 실측 사이즈에 더함으로써, 실제로 받은 옷 사이즈와 유사하게 만들고자함

▼ `total_b_s`를 사용하여 `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s` 데이터 보완 작업을 진행할지의 여부

- `total_b_s` 가 **±1(큼|작음)**이고, `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s` 이 모두 **0(보통)**인 경우 → **4,689**개
- `total_b_s` 가 **0(보통)**이고, `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s` 중에 하나라도 **±1(큼|작음)**인 경우 → **1,539**개

- `total_b_s` 가 **0(보통)**이고, `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s` 도 모두 **0(보통)**인 경우 → **81,074**개
- 데이터 보완 작업을 안하는 방법이 분류성능평가가 **0.0018** 높게 나와서 전처리 작업 없이 `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s`를 **그대로** 사용하고자함
  - 데이터 보완 작업없이 `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s`만 사용해서 분석을 진행할 경우, 분류성능평가에서 accuracy는 **0.7657**임
  - 데이터 보완 작업 후 `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s`를 사용해서 분석을 진행할 경우, 분류성능평가에서 accuracy는 **0.7639**임

- 최종적으로 사이즈 재구성 때 사용하고자 하는 데이터프레임 정보
  - shape : *(87959, 12)*
    - 리뷰 분석을 통해 얻은 **비율값**을 구할 때 사용할 컬럼 목록 : `goodsNo`, `typeClass`, `total_b_s`, `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s`
    - 상품의 각 부위별 사이즈 **간격값**을 구할 때 사용할 컬럼 목록 : `goodsNo`, `size`, `length`, `shoulder`, `chest`, `sleeve`

- 리뷰 분석을 통해 얻은 데이터를 사용하여 **비율값**을 구하기

  ▼ 비율값이 **1**에 가까울수록 다음으로 큰 사이즈의 값과 같다고 판단함
  - 무신사 홈페이지 기준 'L'사이즈 총장의 길이가 60이고 'XL'사이즈 총장의 길이가 65일때, 리뷰에서 모든사람이 총장이 길다고 남겨서 비율값이 1이 나올 경우, 본 프로젝트에서는 실제로 받게되는 'L'사이즈 총장의 길이는 65, 'XL'와 동일하다고 추측함
  - 마찬가지로 비율값이 -1에 가까울수록 다음으로 작은 사이즈의 값과 같다고 봄
  - `goodsNo` 기준으로 각각의 `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s` 값들을 합치고 총 리뷰 데이터 수만큼 나누기(값의 범위 : -1(작음)~1(큼))
    - `goodsNo` 기준으로 각각의 `length_b_s`, `shoulder_b_s`, `chest_b_s`, `sleeve_b_s` 값들을 합치고, `sum_review(총 리뷰 데이터 수)` 컬럼을 추가한 데이터프레임

	goodsNo	length_b_s	shoulder_b_s	chest_b_s	sleeve_b_s	sum_review
0	394585	4	0	0	9	300
1	394608	5	2	1	6	300

    - **비율값**에 대한 최종 데이터프레임 : `b_s_sum_df`
      - shape : *(296, 5)*

	goodsNo	length_b_s	shoulder_b_s	chest_b_s	sleeve_b_s
0	394585	0.013333	0.000000	0.000000	0.03
1	394608	0.016667	0.006667	0.003333	0.02

- 상품의 각 부위별 사이즈 **간격값**을 구하기
  - 상품의 리뷰마다 사이즈가 개별적으로 저장되어있음
  - `goodsNo` 별로 개별 사이즈 정보를 리스트로 묶고, 그 사이즈 정보 리스트들을 한번 더 리스트로 묶어서 **전체 사이즈 정보**를 담음

	goodsNo	size_total
0	394585	[[M, 73.0, 56.0, 58.0, 60.0], [L, 76.0, 60.0, ...
1	394608	[[L, 76.0, 60.0, 61.0, 62.0], [M, 73.0, 56.0, ...
2	404474	[[M, 72.5, 51.0, 55.5, 59.0], [XL, 76.5, 55.0,...

- 상품 별 총장, 어깨, 가슴, 소매 사이즈 값들의 **간격값을** 부위별로 하나의 값으로 구하기
  - 비율값을 옷의 여러 사이즈에 동시에 적용하려면 **하나의 고정값**이 필요함
  - **간격의 최빈값의 평균값(이상치 제거작업 진행 O)**을 사용함
    - 간격값 설정 진행 과정(`length`를 사용하여 설명함)
      - **(총장길이 최대값-최소값)/상품 사이즈 개수**
        - 장점: `length_gap(총장의 간격 값들의 집합)`과 `length_gap_mode(총장의 간격 값들의 최빈값의 집합)`가 여러 개여도 상관없이 계산 적용 가능
        - 단점: 사이즈 정보 끝값(ex.'XS'나 'XXXL')에서 간격 변화가 크게 주어져서 길이의 최대값에서 최소값을 뺄 때 값이 크게 나오는 경우가 발생함

      - **상품별 `length_gap`의 평균값 사용**
        - 장점: `length_gap`이 여러개인 경우 영향 최소화
        - 단점: 여전히 이상치 영향을 크게 받음
      - **상품별 `length_gap_mode`의 평균값 사용**
        - 장점:`length_gap`과 `length_gap_mode`가 여러개인 경우 영향 최소화
        - 단점: 이상치 영향을 `length_gap`보단 적게 받지만 여전히 영향은 있음.
        - 보완점 1 : 간격값들 중 사분위 수에서 **Lower bound(Q1 − 1.5 * IQR)**보다 작고, **Upper bound(Q3 + 1.5 * IQR)**보다 큰 경우를 모두 제외한 후 최빈값 구한 후 평균내기
        - 보완점 1의 문제: 조건을 걸고 나서 간격값이 **Null**로 바뀌는 경우가 생김
          - ex. `shoulder`에서 Upper bound가 **3.85**인데 `shoulder_gap_mode`가 [**4.0**]인 경우
        - 보완점 2: 최빈값을 구한 후, **최빈값의 개수**가 **2개 이상**인 경우, **Lower, Upper bound에 벗어나는 값을 제거한 후 평균내기**
          - **Null**값 없으므로 최종적으로 사용
    - 간격값에 대한 최종 데이터프레임 `gap_mode_not_outlier_df`
      - shape : *(296, 6)*

	goodsNo	size_total	length_gap_mode	shoulder_gap_mode	chest_gap_mode	sleeve_gap_mode
0	394585	[[M, 73.0, 56.0, 58.0, 60.0], [L, 76.0, 60.0, ...	3.0	4.0	2.0	2.0
1	394608	[[L, 76.0, 60.0, 61.0, 62.0], [M, 73.0, 56.0, ...	3.0	4.0	3.0	2.0

💡 **새로 재구성한 사이즈 정보 = 기존 사이즈 정보 + (총장, 어깨, 가슴, 소매의 간격의 최빈값의 평균값) X (리뷰 분석을 통해 얻은 비율값)**

- `gap_mode_not_outlier_d`f와 `b_s_sum_df`를 병합한 후, 새로 재구성한 사이즈 정보를 추가한 최종 데이터프레임
  - shape : *(296, 11)*

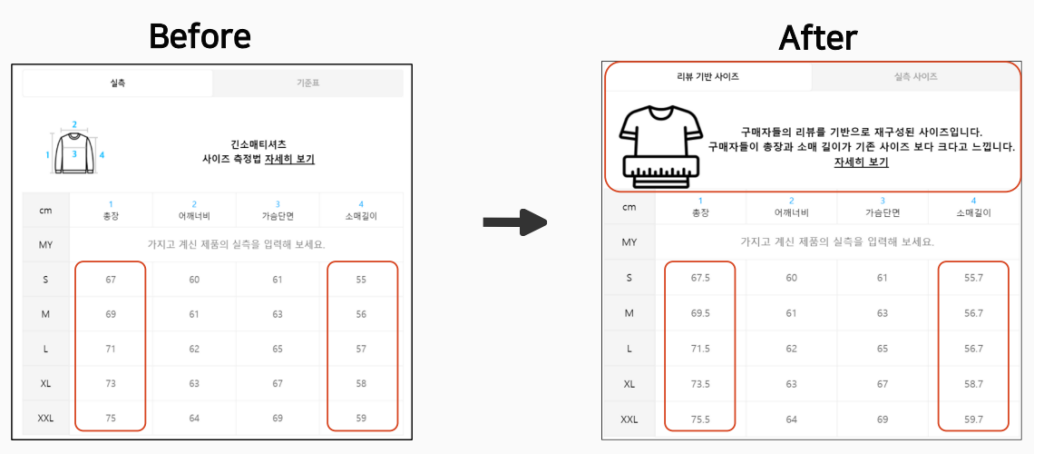
	goodsNo	size_total	length_gap_mode	shoulder_gap_mode	chest_gap_mode	sleeve_gap_mode	length_b_s	shoulder_b_s	chest_b_s	sleeve_b_s	size_new_total
0	394585	[[M, 73.0, 56.0, 58.0, 60.0], [L, 76.0, 60.0, ...	3.0	4.0	2.0	2.0	0.013333	0.000000	0.000000	0.03	[[M, 73.04, 56.0, 58.0, 60.06], [L, 76.04, 60....
1	394608	[[L, 76.0, 60.0, 61.0, 62.0], [M, 73.0, 56.0, ...	3.0	4.0	3.0	2.0	0.016667	0.006667	0.003333	0.02	[[L, 76.05, 60.02666666666666, 61.01, 62.04],...

6. 결과

6-1. 분석 결과

- 리뷰 분석의 정확도 파악
  - **Accuracy** : **0.77**
  - 정답 데이터로 사용한 `typeclass`는 완벽한 정답 데이터로 보기에는 부족하지만, **0.77**이라는 값은 향후 본 프로젝트의 한계점을 보완하면 정확도가 높은 데이터를 구축할 수 있을 것으로 예상함

- 기존의 무신사 스토어 사이트에서 제공하는 실측 사이즈 표를 착안하여, 리뷰를 기반으로 재구성 된 사이즈 표를 통해 사용자에게 더욱 유용한 사이즈 정보를 제공함



6-2. 기대효과 및 활용 방향

- 기대 효과
  - **구매자**
    - 사이즈 정보를 개선함으로 **사용자가 원하는 실제 사이즈의 상품을 구매**하는데 도움이 되어 환불률이 줄어들 것으로 기대 됨
  - **판매자**
    - 환불률이 낮아진다면 **매출 증가**로 이어질 수 있음
    - 환불 배송 건 수가 줄면서 **물류에 투자되는 비용이 감소**될 것으로 기대 됨

- 프로젝트 활용 방향
  - 기존 무신사스토어 사이트 내 **추천 시스템 구현**

        - 다양한 상품 데이터를 수집하고, 사용자의 신체 사이즈 정보를 이용하여, 사용자가 원하는 사이즈를 가진 상품 추천할 수 있음

7. 프로젝트 회고

- 리뷰 개수 **300개** 이상인 상품만 분석 가능함
- 데이터 프레임 형성 시 사이즈 정보 **컬럼 매칭**이 안되어 사용하지 못하는 상품 존재함
- 텍스트 데이터 분석 시 **사전의 단어가 존재해야** 분석이 가능함
- `okt`사용시 동사 기준으로 문장을 잘랐을 때 **동사로 인식을 안되어** 의도대로 문장이 안 잘리는 경우 발생함
- 각 부위에 대해 크고 작음을 언급한 리뷰의 **비율이 적어** 사이즈를 재구성하는데 어려움이 있어 접근하는 방식을 다르게 생각해볼 수 있음
- 분류성능평가에서 타겟 데이터로 `typeClass` 을 사용함
    - 각 부위별 크고 작음에 대한 소비자의 설문조사가 있으면 더 정확한 평가가 가능함
- 사이즈 재구성 계산식은 **한 사이즈 미스**가 난다는 가정하에 진행함
    - 사이즈 교환 정보나 환불 정보가 있으면 계산식 보완이 가능하며 계산식 검증도 가능할 것으로 예상함